

【서지사항】

【서류명】	특허출원서
【권리구분】	특허
【수신처】	특허청장
【참조번호】	0004
【제출일자】	2002.08.28
【국제특허분류】	G06F
【발명의 명칭】	헤더 압축에 의한 패킷 데이터의 송신 방법 및 장치
【발명의 영문명칭】	METHOD AND APPARATUS TRANSMITTING A HEADER COMPRESSED PACKET DATA
【출원인】	
【명칭】	삼성전자 주식회사
【출원인코드】	1-1998-104271-3
【대리인】	
【성명】	이건주
【대리인코드】	9-1998-000339-8
【포괄위임등록번호】	1999-006038-0
【발명자】	
【성명의 국문표기】	나승구
【성명의 영문표기】	NA,Seung Gu
【주민등록번호】	651105-1622315
【우편번호】	449-908
【주소】	경기도 용인시 기흥읍 영덕리 14-11 신일아파트 202동 1702호
【국적】	KR
【발명자】	
【성명의 국문표기】	김용건
【성명의 영문표기】	KIM,Yong Keon
【주민등록번호】	630111-1037024
【우편번호】	449-843
【주소】	경기도 용인시 수지읍 상현리 풍산아파트 101동 1802호
【국적】	KR

【발명자】**【성명의 국문표기】** 전우성**【성명의 영문표기】** JUN, Woo Sung**【주민등록번호】** 740702-1167514**【우편번호】** 463-776**【주소】** 경기도 성남시 분당구 서현동(시범단지) 한양아파트
327-803**【국적】** KR**【공개형태】** 학술단체 서면발표**【공개일자】** 2002.07.12**【취지】** 특허법 제42조의 규정에 의하여 위와 같이 출원합니다. 대
리인 이권
주 (인)**【수수료】****【기본출원료】** 20 면 29,000 원**【가산출원료】** 18 면 18,000 원**【우선권주장료】** 0 건 0 원**【심사청구료】** 0 항 0 원**【합계】** 47,000 원**【첨부서류】** 1. 공지에외적용대상(신규성상실의예외, 출원시의특례)규정
을 적용받 기 위한 증명서류_1통

【요약서】**【요약】**

본 발명은 패킷 데이터 통신 시스템에서 다중 프로토콜 헤더를 가지는 패킷화된 데이터를 송신하고 수신하는 방법에 관한 것으로서, 주기적으로 네트워크 상태를 측정하여 상기 측정 결과에 따라 패킷 전송 주기를 결정한 뒤, 상기 결정된 패킷 전송 주기에서는 압축되지 않은 헤더를 가지는 풀 패킷들을 전송하고 상기 결정된 패킷 전송 주기가 아닌 동안에는 압축된 헤더를 가지는 압축 패킷들을 송신한다. 여기서 패킷 전송 주기는, 최근 네트워크 상태 측정 주기 동안에 측정된 평균 패킷 재전송율에 따라 혼잡 상태이면 1로 정해지고, 무부하 상태이면 정상 상태일 때의 패킷 전송 주기의 2배로 정해진다. 이로써 본 발명은 다른 패킷의 손실에 의한 패킷의 폐기를 예방하여 헤더 압축 기술에 의한 전송효율을 보다 향상시킨다.

【대표도】

도 6

【색인어】

packet data, header compression, TCP/IP

【명세서】

【발명의 명칭】

헤더 압축에 의한 패킷 데이터의 송신 방법 및 장치{METHOD AND APPARATUS
TRANSMITTING A HEADER COMPRESSED PACKET DATA}

【도면의 간단한 설명】

도 1은 본 발명이 적용될 수 있는 무선 패킷 데이터 전송 시스템의 구성을 도시한 블록도.

도 2는 TCP/IP에 의한 패킷의 포맷.

도 3은 압축된 TCP/IP 헤더를 가지는 패킷 포맷의 일 예.

도 4는 본 발명에 따른 헤더 압축 기술을 사용하는 통신 노드의 구성을 도시한 블록도.

도 5는 통상의 패킷 전송을 예시한 타이밍도.

도 6은 본 발명에 따라 분류된 네트워크의 세 가지 상태들을 나타낸 상태 천이도.

도 7은 본 발명에 따라 적응적인 주기를 이용한 패킷 전송을 예시한 타이밍도.

도 8은 본 발명에 따른 패킷 전송 동작을 나타낸 흐름도.

도 9는 상기 도 8에 나타낸 동작을 나타낸 C언어 알고리즘.

【발명의 상세한 설명】

【발명의 목적】

【발명이 속하는 기술분야 및 그 분야의 종래기술】

- <10> 본 발명은 패킷 데이터 통신 시스템에 관한 것으로서, 특히 다중 프로토콜 헤더를 가지는 패킷화된 데이터(Packetized Data)를 송신하고 수신하는 방법에 관한 것이다.
- <11> 인터넷을 통해 데이터를 전송하기 위한 프로토콜로서, 인터넷 표준화 단체인 IETF(Internet Engineering Task Force)에 의해 발행된 RFC(Request For Comments) 793 에 전송제어 프로토콜(Transmission Control Protocol)이 규정되어 있다. TCP는 인터넷 을 통해 노드들간에 데이터를 패킷의 형태로 전송하기 위해 사용된다. 이러한 TCP는, 네 트워크 계층의 인터넷 프로토콜(Internet Protocol: IP)에 대한 상위에서 트랜스포트 계 층(Transport Layer)의 프로토콜로서 동작하기 때문에, 통상의 경우 TCP/IP라 표기한다.
- <12> 상기 TCP/IP를 포함한 전형적인 데이터 전송 기술에서, 패킷화된 데이터 (Packetized Data)는 네트워크 프로토콜 기능을 제어하기 위해 사용되는 적어도 하나의 헤더를 포함한다. 헤더는 통신 채널에서 사용 가능한 대역폭을 소모시키는 한 원인이 되 기 때문에 이를 압축하여 대역폭의 사용을 최적화하기 위한 많은 기술들이 연구되고 있 다. 이는 케이블을 사용하는 유선통신 시스템에서보다는, 음성 및/또는 데이터를 무선 주파수(Radio Frequency: RF)를 통해 전송하도록 설계된 무선통신 시스템에서 더 중요한 논점이 된다.
- <13> 특히 TCP/IP 헤더를 압축하기 위하여 반 제이콥슨(Van Jacobson)에 의하여 제안된 방법은 TCP/IP 패킷에서 여분의 정적 정보(redundant static information)를 제거하고

동적 정보(changing information)를 예측할 수 있는 차이값(differences)을 전송한다. 이러한 압축에 대해서 상술하면, 두 노드들 중 하나의 노드(이하 "송신 노드"라 한다.)는 다른 노드(이하 "수신 노드"라 한다.)에게 압축되지 않은 헤더를 가지는 패킷들(이하 "풀 패킷(Full Packet)"이라 한다.)을 주기적으로 전송한다. 이때 풀 패킷들 사이에는 정적 정보가 제거되고 동적 정보의 차이값들을 가지도록 압축된 헤더를 가지는 패킷들(이하 "압축 패킷(Compressed Packet)"이라 한다.)이 전송된다. 수신 노드는 압축 패킷의 수신시, 이전에 수신된 헤더를 참고하여 압축 패킷의 헤더를 복원한다.

<14> 그런데 만약 통신 채널의 상태가 불량하거나 네트워크 상태가 혼잡(congestion)하여 연속된 패킷들 중 하나가 유실 또는 손상된 경우, 수신 노드는 그 다음으로 수신되는 압축 패킷들의 헤더들을 복원할 수 없다. 게다가 TCP/IP에 따르면 수신 노드는 압축 패킷들에 대해서는 응답(Acknowledge)을 보내지 않기 때문에 수신 노드는 풀 패킷을 수신한 이후에야 상기 유실 또는 손상된 패킷의 재전송을 요구할 수 있게 된다. 결과적으로 손실이 발생한 이후 풀 패킷이 도착하기 전까지 수신되는 압축 패킷들은 폐기될 수밖에 없다.

<15> 이상에서 설명한 바와 같이 TCP/IP 패킷의 헤더를 압축하여 전송하는 기술은 정적 정보에 관련된 문제점에 대한 단순한 해결책이 되는 반면 무선 환경(Radio Frequency environments)에서 중요한 문제점들의 원인이 된다. 특히 무선 환경에서는 빈번한 오류들이 발생하게 되는데, 이러한 오류들이 발생되면 데이터 전송은 손실 패킷이 다시 전송될 때까지 중단될 수밖에 없고 이로써 오히려 전송효율이 저하된다는 문제점이 발생하였다.

【발명이 이루고자 하는 기술적 과제】

- <16> 따라서 상기한 바와 같이 동작되는 종래 기술의 문제점을 해결하기 위하여 창안된 본 발명은 헤더 압축에 의한 패킷 데이터 전송 시스템에서 전송효율을 증대시키는 방법 및 장치를 제공한다.
- <17> 본 발명은 무선통신시스템을 포함하여 TCP/IP를 사용하는 패킷 데이터 전송 시스템에서 전송효율을 증대시키는 방법 및 장치를 제공한다.
- <18> 본 발명은 헤더 압축에 의한 패킷 데이터 전송 시스템에서 패킷의 손실로 인한 전송효율의 저하를 방지하는 방법 및 장치를 제공한다.
- <19> 본 발명은 헤더 압축에 의한 패킷 데이터 전송 시스템에서 네트워크 상태에 따라 압축되지 않은 헤더를 가지는 패킷을 전송하는 방법 및 장치를 제공한다.
- <20> 상기한 바와 같은 목적을 달성하기 위하여 창안된 본 발명의 실시예는, 패킷 통신 네트워크에서 패킷 데이터를 송신하는 방법에 있어서,
- <21> 주기적으로 네트워크 상태를 측정하고, 상기 측정 결과에 따라 패킷 전송 주기를 결정하는 과정과,
- <22> 상기 결정된 전송 주기에서는 압축되지 않은 헤더를 가지는 풀 패킷들을 전송하고 상기 결정된 전송 주기가 아닌 동안에는 압축된 헤더를 가지는 압축 패킷들을 송신하는 과정을 포함한다.
- <23> 본 발명의 다른 실시예는, 패킷 통신 네트워크에서 패킷 데이터를 송신하는 장치에 있어서,

- <24> 주기적으로 네트워크 상태를 측정하고, 상기 측정 결과에 따라 패킷 전송 주기를 결정하는 제어기와,
- <25> 상기 결정된 전송 주기에서는 압축되지 않은 헤더를 가지는 폴 패킷들을 전송하고 상기 결정된 전송 주기가 아닌 동안에는 압축된 헤더를 가지는 압축 패킷들을 송신하는 송신기를 포함한다.

【발명의 구성 및 작용】

- <26> 이하 첨부된 도면을 참조하여 본 발명의 바람직한 실시예에 대한 동작 원리를 상세히 설명한다. 하기에서 본 발명을 설명함에 있어 관련된 공지 기능 또는 구성에 대한 구체적인 설명이 본 발명의 요지를 불필요하게 흐릴 수 있다고 판단되는 경우에는 그 상세한 설명을 생략할 것이다. 그리고 후술되는 용어들은 본 발명에서의 기능을 고려하여 정의된 용어들로서 이는 사용자, 운용자의 의도 또는 관례 등에 따라 달라질 수 있다. 그러므로 그 정의는 본 명세서 전반에 걸친 내용을 토대로 내려져야 할 것이다.
- <27> 후술되는 본 발명은 헤더 압축을 사용하는 패킷 데이터 전송 시스템에서 네트워크 상태에 따라 압축되지 않은 헤더를 가지는 패킷 데이터를 송신하는 주기를 결정하는 것이다. 특히 본 발명은 패킷의 재전송율에 따라 네트워크 상태를 측정한다.
- <28> 고속 무선 패킷 데이터 통신에 대한 사용자의 요구가 증대됨에 따라 이동통신 네트워크를 인터넷 등의 패킷 네트워크와 연동시키는 다양한 방안들이 시도되고 있다. 이러한 시도 중의 하나로 부호분할다중접속(Code Division Multiple Access)2000 1xEVDO(Evolution in Data Only) 시스템은 패킷 데이터 서비스 노드(Packet Data

Serving Node)라는 게이트웨이를 이용하여 인터넷에 접속한다. 이러한 패킷 데이터 서비스 노드는 인터넷에 접속하는 컴퓨터와 이동 단말기 사이에서 인터넷 프로토콜에 의한 패킷 데이터들을 중계한다.

<29> 도 1은 본 발명이 적용될 수 있는 무선 패킷 데이터 전송 시스템의 구성을 도시한 블록도이다.

<30> 상기 도 1을 참조하면, 패킷 데이터 전송 시스템은 무선 채널을 통해 이동 단말기(10)를 서비스하는 기지국 송수신기(Base Transceiver Subsystem: BTS)(12)와 상기 기지국 송수신기(12)를 제어하는 기지국 제어기(Base Station Controller: BSC)(14)와 상기 기지국 제어기(14)를 패킷 데이터 서비스 노드(PDSN)(18)를 통해 네트워크(30)로 연결하는 GAN(Global ATM(Asynchronous Transfer Mode) Network) 교환기(16)로 구성된다. 이러한 구성을 통해 상기 이동 단말기(10)는 네트워크(30)에 접속하는 컴퓨터(20)와 패킷 데이터를 교환한다.

<31> 여기서 상기 이동 단말기(10)는 셀룰러 전화기(Cellular Phone)나 PCS(Personal Communications Service) 전화기에 접속된 노트북 또는 인터넷 통신이 가능한 PDA(Personal Digital Assistant) 전화기나 IMT(International Mobile Telecommunication)-2000 단말기 등이며, 상기 컴퓨터(20)는 웹 서버(Web Server), FTP(File Transfer Protocol) 서버 등이다. 예를 들어 FTP 서버(20)는 사용자에게 의해 요구된 파일을 패킷 데이터의 형태로 이동 단말기(10)에게 제공한다. 이러한 경우 상기 FTP 서버는 송신 노드가 되고 상기 이동 단말기(10)는 수신 노드가 된다.

<32> 상기 도 1은 이동 단말기로서의 송신 노드와 FTP 서버로서의 수신 노드 및 이동 단말기와 FTP 서버간을 연결하는 다수의 장치들을 개시하고 있지만, 본 발명이 이러한 구

성에 한정되지 않음은 물론이다. 즉 본 발명은 유사한 기술적 배경을 가지는 여타의 데이터 전송 시스템에도 본 발명의 범위를 크게 벗어나지 아니하는 범위에서 약간의 변형으로 적용 가능하며, 이는 본 발명의 분야에서 숙련된 기술적 지식을 가진 자의 판단으로 가능할 것이다.

- <33> 상기와 같은 구성에 있어서 송신 노드는 전송하고자 하는 파일들을 분할한 패킷들을 순서대로 수신 노드에게 송신한다. 인터넷 프로토콜(Internet Protocol: IP)에 기반한 전송 제어 프로토콜(Transmission Control Protocol: TCP)(이하 "TCP/IP"라 한다.)를 사용하는 통신에 있어서, 송신 노드로부터 송신되는 패킷의 포맷은 도 2에 나타내었다. 상기 도 2를 참조하면, TCP/IP 헤더는 IP 헤더 및 TCP 헤더를 포함한다.
- <34> 상기 IP 헤더는 일반적으로 여분의 정적 정보(음영으로 표시함)를 포함한다. 4비트의 프로토콜 버전(Protocol Version)은 인터넷 헤더의 포맷을 나타낸다. 이하 RFC 791에 개시된 버전 4 포맷을 설명한다. 4비트의 헤더 길이(Header Length)는 인터넷 헤더의 길이이며 데이터의 시작을 지시한다. 서비스 유형(Type Of Service)은 지연(Delay)과 신뢰성(Reliability) 및 효율성(Throughput)의 측면에서 원하는 서비스의 품질을 나타내는 8비트의 정보이다. 16비트의 전체 길이(Total Length)는 옥텟 단위로 측정된 패킷(헤더 및 데이터)의 길이이다. 패킷 식별자(Packet Identifier)는 데이터그램의 조각들(fragments)을 조립하기 위해 송신 노드에서 할당한 16비트의 식별값이다. 각각 1비트인 3개의 플래그들 중 첫 번째 예비된 비트는 0으로 설정되며, 두 번째 비트인 DF는 조각(Fragment)인지의 여부를 나타내고, 세 번째 비트인 MF는 마지막 조각인지의 여부를 나타낸다. 13비트의 조각 오프셋(Fragment Offset)은 해당 조각이 데이터그램의 어느 위치에 속하는지를 나타낸다. 유효시간(Time To Live: TTL)은 해당 데이터그램이 잔류할 수 있

는 최대 시간을 8비트로 나타낸다. 8비트의 프로토콜 식별자(Protocol Identifier)는 데이터그램의 데이터 부분에서 사용되는 프로토콜(여기서는 TCP)을 나타낸다. 16비트의 헤더 체크섬(Header Checksum)은 헤더만의 오류정정 정보이다. 근원지 주소와 목적지 주소는 각각 근원지와 목적지의 32비트 IP 주소를 나타낸다. 이들 중 변화하는 정보는 전체 길이와 패킷 식별자와 헤더 체크섬이다.

<35> 또한 상기 TCP 헤더에 대해 설명하면, 근원지 포트(Source Port)와 목적지 포트(Destination Port)는 각각 근원지와 목적지의 16비트로서 포트번호를 나타낸다. 32비트의 시퀀스 번호(Sequence Number)는 첫 번째 데이터 옥텟의 시퀀스 번호를 나타낸다. 32비트의 응답 번호(Acknowledge Number)는 송신 노드에서 수신하기를 기대하는 다음 시퀀스 번호이다. 4비트의 데이터 오프셋은 TCP 헤더에서 32비트 워드의 길이로서 헤더의 길이를 나타낸다. 6비트의 예비된 필드는 0으로 설정되어야 한다. 제어 비트들(Control Bits)은 표준화된 TCP 응답(Acknowledge: ACK)의 경우에 응답(Acknowledge)의 유형을 결정하는데 사용되는 6개의 비트들(URG, ACK, PSH, RST, SYN, FIN)이다. 상기 제어 비트들의 의미는 하기에 나타낸 바와 같다.

<36> URG(Urgent Pointer) : 긴급 지시자(Urgent Pointer) 필드의 유효 여부를 나타냄.

<37> ACK(Acknowledge) : 패킷이 응답을 구성하는지의 여부를 나타냄.

<38> PSH(Push) : "푸쉬(Push)" 기능이 요구되었는지를 나타냄.

<39> RST(Reset) : 접속 리셋이 요구되었는지를 나타냄.

<40> SYN(Synchronization) : 시퀀스 번호들을 동기화함.

<41> FIN(Final) : 송신 노드에서 더 이상 전송할 데이터가 없음을 나타냄.

- <42> 16비트의 윈도우 크기(Window Size)는 송신 노드에서 수용할 수 있는 시퀀스 번호의 최대 크기를 나타낸다. 16비트의 TCP 체크섬은 헤더와 데이터의 체크섬이다. 16비트의 긴급 지시자(Urgent Pointer)는 이어지는 긴급 데이터의 시퀀스 번호를 나타낸다. 상기 TCP 헤더에서 변화하는 정보는 근원지 및 목적지 포트와 데이터 오프셋 및 제어 비트들 중 일부를 제외한 나머지이며, 정적 정보는 음영으로 표시하였다.
- <43> 이상의 구성에서 IP 헤더는 20바이트이고 TCP 헤더 또한 20바이트이지만 TCP 접속이 이루어진 이후 전체 40바이트의 오버헤드에서 음영으로 표시된 약 20바이트의 정적정보는 변하지 않는다. 앞서 언급한 바와 같이 매 패킷을 전송할 때마다 40바이트의 오버헤드 전체를 전송하는 것은 매우 비효율적이므로, 전체(압축되지 않은) 헤더를 가지는 패킷들(이하 "풀 패킷"이라 한다.) 사이에는 정적 정보가 제거되고 동적 정보의 차이값들을 가지도록 압축된 헤더를 가지는 패킷들(이하 "압축 패킷"이라 한다.)이 전송된다.
- <44> 헤더 압축 기술이 사용되는 경우에 풀 패킷의 헤더는 압축된 헤더를 복원하는데 사용되는 컨텍스트(Context)를 나타내는 컨텍스트 식별자(Context Identifier: CID)를 더 포함한다. 컨텍스트는 송신 노드에서 헤더를 압축하고 수신 노드에서 헤더를 복원(decompress)하는 상태를 나타낸다.
- <45> 도 3은 IETF에 의해 발행된 RFC2507에 개시되어 있는 압축된 TCP/IP 헤더를 가지는 패킷의 포맷을 나타낸 것으로서, 도시한 바와 같이 압축 패킷은 도 2에서 음영으로 표시한 정적 정보를 포함하지 않는다. 수신 노드는 압축 패킷에 포함된 컨텍스트 식별자에 의해 특정화되는 TCP/IP 헤더의 내용에 따라 압축 헤더를 복원한다. TCP 체크섬은 종단 간(End to End) 데이터의 무결성 확인을 위한 것이므로 변경되지 않은 채로 전송된다.

- <46> TCP/IP 헤더에서 윈도우 크기, 응답 번호, 시퀀스 번호, 패킷 식별자는 순서대로 송신된 연속하는 패킷들간의 변화량이 일정하다고 기대된다. 이하 이러한 데이터를 제공하는 필드를 "델타 필드"라 한다. 델타 필드의 변화량이 일정한 경우에 도 3에 나타낸 바와 같이 기본 압축 헤더는 델타 필드 내의 데이터를 포함하지 않는다. 전송한 바와 같이 이전 수신한 패킷으로부터 추출한 풀 헤더를 저장하고 있는 수신 노드는 TCP/IP 헤더의 델타 필드에 기록된 데이터에 이전에 구한 일정 변화량을 각각 가산하여 이후 연속하여 수신되는 압축 패킷의 델타 필드를 복원한다.
- <47> 그런데 델타 필드의 변화량이 모든 패킷들간에 항상 일정한 것은 아니다. 변화량에 변화가 생기게 되는 경우 송신 노드는 수신 노드에게 변경된 차이값을 통보하여야 한다. 수신 노드는 저장된 풀 헤더의 내용과 통보된 차이값을 참조하여, 이후 수신되는 압축 패킷의 델타 필드를 복원할 수 있다. 이런 목적으로 도 3에 나타낸 압축 패킷은 델타 필드의 변화량이 변경되었는지의 여부를 나타내는 변경 마스크 정보, 즉 S, A, W, I 플래그들을 가진다.
- <48> 상기 S, A, W, I 플래그들은 각각 TCP/IP 헤더의 시퀀스 번호, 응답 번호, 윈도우 크기, 패킷 식별자의 변화 여부를 나타낸다. 델타 필드의 변화량이 변경된 경우에, 압축 패킷에는 최종 송신된 패킷의 해당 필드와의 차이값을 나타내는 해당 필드가 부가된다. 실제로, 시퀀스 번호의 변화량이 변경된 경우에 S 플래그가 '1'로 설정됨과 동시에 시퀀스 번호의 차이값을 나타내는 델타 시퀀스 필드가 압축 패킷에 부가된다. 마찬가지로 윈도우 크기나 응답 번호나 패킷 식별자의 변화량이 변경된 경우에, W, A, I 플래그가 '1'로 설정되고 해당 차이값을 나타내는 필드가 압축 패킷에 부가된다. 수신 노드는 압축

패킷을 수신하면 최종 수신한 패킷의 해당 필드에 상기 압축 패킷의 차이값을 적용하여 완전한 값으로 복원한다.

<49> 또한 U 플래그와 P 플래그는 각각 TCP/IP 헤더의 제어 비트들 중 URG 비트와 PSH 비트를 복사한다. C 플래그는 컨텍스트 식별자(Context Identifier: CID)가 항상 존재하기 때문에 제거될 수 있다. 이와 같이 압축된 헤더의 크기는 9바이트 이하가 되어 도 2에 나타난 압축되지 않은 헤더에 비하여 반 이상 줄어들게 된다.

<50> 도 4는 본 발명에 따른 헤더 압축 기술을 사용하는 통신 노드의 구성을 도시한 블록도이다. 이 구성은 도 1에 나타난 이동 단말기(10) 및 FTP 서버(20)에 적용된다.

<51> 상기 도 4를 참조하면, 통신 노드는 송신부(100)와 수신부(110)와 네트워크 인터페이스(140) 및 이들을 상호 연결하는 버스(130)를 포함한다. 수신부(110)는 송신측 노드로부터 송신된 풀 패킷이나 압축 패킷을 통신 회선을 통해 수신하여, 그 패킷을 분석한다. 송신부(100)는 전송하고자 하는 데이터를 풀 패킷이나 압축 패킷으로 변환하여 통신 회선을 통해 수신측 노드로 송신한다.

<52> 구체적으로 설명하면, 송신부(100)의 패킷 생성기(Packet Generator)(102)는 전송하고자 하는 데이터에 TCP 헤더와 IP 헤더를 부가한 IP 패킷을 생성하며, 압축기(Compressor)(104)는 상기 IP 패킷을 압축되지 않은 헤더를 포함하는 풀 패킷 또는 압축된 헤더를 포함하는 압축 패킷으로 변환한다. 상기 IP 패킷은 다음 전송될 압축 패킷의 생성에 이용될 수 있도록 메모리(105)에 저장된다.

<53> 프레임어(Framer)(108)는 상기 풀 패킷 또는 상기 압축 패킷을 버스(130)를 통해 네트워크 인터페이스(140)로 전달한다. 네트워크 인터페이스(140)는 상기 풀 패킷 또는

상기 압축 패킷을 통신 회선을 통해 해당하는 수신 노드로 전송한다. 여기서 상기 압축기(104)는 미리 정해지는 주기에서는 풀 패킷을 출력하고 다른 경우에는 압축 패킷을 출력한다. 상기 압축기(104)가 풀 패킷을 전송하는 주기는 제어기(106)가 측정된 네트워크 상태에 따라 결정된다. 상기 제어기(106)에 의한 처리에 대해서는 추후 상술할 것이다.

<54> 수신부(110)의 디프레이머(Deframer)(118)는 통신 회선을 통해 송신 노드로부터 풀 패킷 또는 압축 패킷을 수신한다. 복원기(Decompressor)(114)는 상기 풀 패킷 또는 상기 압축 패킷을 IP 패킷으로 변환하여 패킷 분석기(Packet Analyzer)(112)로 전달한다. 이때 상기 복원기(114)는 상기 변환된 IP 패킷의 헤더를 메모리(116)에 저장하고, 다음에 수신되는 압축 패킷의 헤더를 복원하는데 이용한다. 도 3에 나타낸 바와 같이 압축 헤더는 컨텍스트 식별자를 포함한다. 따라서 상기 복원기(114)는 상기 컨텍스트 식별자에 의해 특정화되는 TCP/IP 헤더의 내용에 따라 압축 헤더를 복원한다.

<55> 도 5는 통상의 패킷 전송을 예시한 타이밍도이다.

<56> 송신 노드의 패킷 생성기에서 IP 패킷(1)이 발생하면, 송신 노드의 압축기는 상기 IP 패킷의 TCP/IP 헤더를 내부 메모리에 저장한다. 동시에 상기 IP 패킷에 컨텍스트 식별자 값을 설정하여 풀 패킷(1)을 생성하고 이를 수신 노드에 송신한다. 수신 노드의 복원기는 상기 풀 패킷(1)으로부터 컨텍스트 식별자 값을 추출하여 TCP/IP 헤더를 복원하고, 상기 복원된 헤더를 포함하는 IP 패킷(1)을 분석한다. 이때 상기 복원된 헤더는 수신 노드의 내부 메모리에 저장된다.

<57> 다음으로 송신 노드의 압축기는 다음으로 발생된 IP 패킷(2)의 TCP/IP 헤더를 압축하고, 상기 압축된 헤더를 포함하는 압축 패킷(2)을 수신 노드에 송신한다. 상기 압축 패킷내의 압축 헤더에는 이전 IP 패킷(1)의 델타 필드 값과 현재 IP 패킷(2)의 델타 필드

드 값과의 차이값을 포함하는 동시에 해당 델타 필드의 변경 여부를 나타내는 플래그가 '1'로 설정된다. 상기 압축 패킷(2)을 수신한 수신 노드의 복원기는 내부 메모리에 저장된 이전 IP 패킷(1)의 TCP/IP 헤더내의 델타 필드 값에 상기 압축 패킷(2)에 통보된 차이값을 부가함으로써 IP 패킷(2)의 델타 필드를 구한다. 그리고 수신 노드의 복원기는 상기 구해진 델타 필드 및 상기 IP 패킷(1)으로부터 추출되어 저장된 TCP/IP 헤더의 정적 필드를 모두 포함하는 TCP/IP 헤더와 상기 압축 패킷(2)의 페이로드를 포함하는 IP 패킷(2)을 분석한다. 상기 IP 패킷(2)을 복원할 때 참조되는 TCP/IP 헤더(이 경우 이전 IP 패킷(1)의 TCP/IP 헤더)는 상기 압축 패킷(2)의 컨텍스트 식별자에 의해 특정화된다. 또한 상기 IP 패킷(2)의 TCP/IP 헤더는 내부 메모리에 저장된다.

<58> 마찬가지로 송신 노드의 압축기는 다음으로 발생된 IP 패킷(3)을 압축 패킷(3)으로 변환하여 수신 노드에 송신한다. 그러나 네트워크 상태의 혼잡 등의 어떠한 이유로 인하여 상기 압축 패킷(3)은 통신 채널 상에서 유실된다. 또는 수신 노드에서 상기 압축 패킷(3)의 체크섬 분석 결과 손상된 것으로 판단한다. 그 결과 수신 노드의 복원기는 다음으로 수신된 압축 패킷(4)의 델타 필드를 복원하는 것이 불가능하다. 따라서 수신 노드에서 다음 풀 패킷(6)이 수신될 때까지 수신되는 패킷들, 즉 압축 패킷들(4,5,6)은 폐기될 수밖에 없다. 특히 송신 노드와 수신 노드가 무선 링크로 접속된 경우에 패킷들은 빈번하게 유실된다. 이러한 유실이 발생하면 수신 노드에서는 몇 개의 패킷들이 자주 폐기된다. 이는 데이터 전송율을 심각하게 저하시킨다.

<59> 송신 노드는 전송한 패킷에 대하여 중복 응답(Duplicate ACK)을 받으면 그 패킷을 즉시 재전송한다. 그렇지 않은 경우 일정한 타임아웃 시간 동안 응답(ACK)을 받지 못하면 그 패킷을 재전송한다. 그런데 수신 노드는 압축 패킷들에 대해서

는 응답을 전송하지 않기 때문에 손실이 발생한 패킷을 수신하기 위해서는 송신 노드에서 상기 타임아웃 시간이 만기될 때까지 기다려야 한다. 게다가 송신 노드는 패킷 손실이 발생한 경우 수신 윈도우의 크기를 줄여 데이터율을 감소시키기 때문에, 결국 데이터 전송율은 매우 저하될 수밖에 없다.

<60> 이러한 문제점을 해결하기 위한 기술로서 IETF에 의해 발행된 RFC 2507은 미리 정해진 주기에 따라 풀 패킷을 주기적으로 송신한다. 그러나 이러한 기술 또한 풀 패킷의 송신 주기가 길어지면 폐기되는 패킷들의 수가 증가하고, 풀 패킷의 송신 주기가 짧아지면 오버헤드가 큰 풀 패킷의 송신으로 인하여 통신 효율이 저하된다는 다른 문제점을 야기한다.

<61> 따라서 본 발명은 패킷 재전송의 발생량에 따라 적응적으로 동작하는 알고리즘을 제안한다. 즉, 패킷 손실이 빈번할 경우에는 헤더 압축을 사용하지 않는 경우가 오히려 더 양호한 성능을 보인다. 따라서 만약 패킷 손실이 빈번하지 않다면 헤더 압축을 사용하고 패킷 손실이 빈번하면 헤더 압축을 사용하여 앞서 언급한 문제점을 해소한다.

<62> 이를 위해서 네트워크 상태를 혼잡(Congested), 부하(Loaded), 무부하(Unloaded)의 세 가지 상태로 분류한다. 바람직하게, 상기 네트워크 상태는 일정한 시간 동안 송신 노드에서 발생한 재전송 패킷들의 개수로부터 계산된 패킷 재전송률에 따라 판단될 수 있다. 패킷 재전송율에 의해 네트워크 상태가 결정되면 각 네트워크 상태에 따라 압축되지 않은 풀 패킷을 보내는 전송 주기를 결정한다.

- <63> 도 6은 본 발명에 따라 분류된 네트워크의 세 가지 상태들을 나타낸 것이다. 여기서 R은 패킷 재전송율을 나타내며 주기는 압축하지 않은 패킷을 전송하는 주기를 나타낸다.
- <64> 도시한 바와 같이, R이 상위 임계값(TH_{HIGH})보다 크거나 같으면 혼잡 상태(200)로, R이 하위 임계값(TH_{LOW})보다 작거나 같으면 무부하 상태(220)로, R이 상위 임계값보다 작고 하위 임계값보다 크면 정상 상태(210)로 분류한다. 바람직한 일 예로서, 상위 임계값은 0.10이고 하위 임계값은 0.05이다. 네트워크가 혼잡 상태(200)이면 풀 패킷의 전송 주기를 짧게 하는 것이 재전송 패킷을 줄일 수 있고 무부하 상태(220)이면 주기를 길게 하는 것이 높은 압축성능을 달성할 수 있다. 정상 상태(210)에서의 주기를 n이라고 한다면, 혼잡 상태(200)에서는 주기를 1로 하여 매번 풀 패킷들을 전송하고, 무부하 상태(220)에서는 주기를 $2n$ 으로 한다.
- <65> 도 7은 본 발명에 따라 적응적인 주기를 이용한 패킷 전송을 예시한 타이밍도이다. 여기서 정상 상태에서의 주기는 3인 것으로 한다. 즉 정상 상태에서 풀 패킷은 매 3번째마다 전송된다.
- <66> 상기 도 7을 참조하면, 무부하 상태인 구간에서 풀 패킷(1)이 전송되고 5개의 압축 패킷들(2 내지 6)이 전송된 이후 다음 풀 패킷(7)이 전송된다. 정상 상태인 구간에서는 2개의 압축 패킷들(8,9)이 전송된 이후 다음 풀 패킷(10)이 전송된다. 혼잡 상태인 구간에서는 압축 패킷은 전송되지 않으며 매번 풀 패킷들(11,12,13)이 전송된다.
- <67> 도 8은 본 발명에 따른 패킷 전송 동작을 나타낸 흐름도이며, 도 9에는 상기 도 8에 나타낸 동작을 C언어 알고리즘으로 나타내었다.

- <68> 이하 상기 도 8과 앞서 언급한 도 4를 참조하여 본 발명에 따른 동작을 상세히 설명한다.
- <69> 과정(S110)에서 송신 노드의 제어기는 미리 정해지는 네트워크 상태 측정 주기에 도달하였는지를 판단한다. 일 예로서, 상기 계산 주기는 RFC1889에 의해 정해진 RTP(Real-time Transmission Protocol)에서 상태정보를 전송하는 RTCP(Real-time Transmission Control Protocol) 피드백 주기인 5초이다. 측정 주기에 도달하였으면, 과정(S120)에서는 최근 측정 주기 동안에 전송한 패킷들의 개수에 대해 재전송된 패킷들의 개수의 비율로서 패킷 재전송율(R)을 계산한다. 과정(S130)에서는 상기 계산된 패킷 재전송율을 이전에 계산된 패킷 재전송율들에 누적하여 평균 패킷 재전송율(AR)을 계산한다. 하기에서 상기 과정(S130)을 보다 상세히 설명한다.
- <70> 매 측정 주기마다 계산되는 패킷 재전송율은 네트워크 상태에 따라 현격하게 변화하게 될 수 있다. 이러한 진동(Oscillation) 현상에 의한 영향을 가급적 제거하여 실제로 적용 가능한 평균 패킷 재전송율(AR)을 계산하기 위하여 저역통과필터링(Low Pass Filtering) 공식을 이용한다. 즉 평균 패킷 재전송율(AR)은 하기의 <수학식 1>과 같다.
- <71> 【수학식 1】 $AR(i) = (1 - \alpha) * AR(i-1) + \alpha * R$
- <72> 여기서 R은 현재 측정 주기에서 계산된 패킷 재전송율이며 AR은 평균 패킷 재전송율이고 i는 측정 주기를 식별하는 인덱스이다. 즉 AR(i-1)은 이전 측정 주기의 평균 패킷 재전송율이며 AR(i)는 현재 측정 주기의 패킷 재전송율이다. 또한 α 는 0과 1 사이의 값을 가지는 가중치로서 이 값이 클수록 가장 최근에 계산된 패킷 재전송율(R)의 영향

이 크게 작용한다. 바람직하게 α 는 0.5와 0.9 사이의 값으로 설정되는 것이 바람직하다. 일 예로서 α 는 0.75로 설정된다.

<73> 과정(S140)에서 상기 평균 패킷 재전송율은 미리 정해지는 상위 임계값(TH_{HIGH}) 및 미리 정해지는 하위 임계값(TH_{LOW})과 각각 비교된다. 비교결과 상기 상위 임계값보다 크거나 같으면 과정(S150)으로 진행하고, 상기 하위 임계값보다 작거나 같으면 과정(S170)으로 진행하며, 그렇지 않으면 과정(S160)으로 진행한다. 일 예로서 상기 상위 임계값은 0.10이고 상기 하위 임계값은 0.05이다. 여기서 0.10은 10%의 평균 패킷 재전송율을 의미하며, 0.05는 5%의 평균 패킷 재전송율을 의미한다.

<74> 과정(S150)에서 네트워크 상태는 혼잡 상태인 것으로 결정되며 풀 패킷의 전송 주기는 1이 되고, 과정(S160)에서 네트워크 상태는 정상 상태인 것으로 결정되며 풀 패킷의 전송 주기는 n 이 되며, 과정(S170)에서 네트워크 상태는 무부하 상태인 것으로 결정되며 풀 패킷의 전송 주기는 $2n$ 이 된다. 여기서 언급한 전송 주기는 네트워크 전송 선로의 품질 및 신뢰성에 따라 변경될 수 있음은 물론이다. 예를 들어 혼잡 상태에서도 풀 패킷의 전송 주기는 2 또는 3이 될 수 있다. 또한 무부하 상태에서도 풀 패킷의 전송 주기는 n 보다 큰 어떤 다른 값이 될 수 있다. 이상과 같이 매 측정 주기마다 결정된 전송 주기는 제어기로부터 압축기로 제공된다.

<75> 그러면 과정(S180)에서 압축기는 상기 결정된 전송 주기에 따라 패킷을 전송한다. 즉, 무부하 상태이면 패킷 손실이 거의 없는 상황이므로 $2n$ 번째마다 풀 패킷을 전송하고, 정상 상태이면 패킷 손실이 약간 발생하는 상황이므로 n 번째마다 풀 패킷을 전송하며, 혼잡 상태이면 패킷 손실이 빈번하게 발생하는 상황이므로 매번 풀 패킷을 전송한다. 풀

패킷의 전송 주기가 아닌 때에는 압축 패킷들을 전송한다. 여기서 압축 패킷들을 생성하는 방법은 전술한 바와 같다.

<76> 본 발명은 풀 패킷의 전송 주기에 관련된 것이며, 수신 노드에서는 수신된 패킷이 풀 패킷인지 또는 압축 패킷인지를 식별하여 분석하게 된다. 이러한 수신 노드에서의 동작은 통상의 동작과 동일하므로 그 상세한 설명을 생략하기로 한다.

<77> 한편 본 발명의 상세한 설명에서는 구체적인 실시예에 관해 설명하였으나, 본 발명의 범위에서 벗어나지 않는 한도 내에서 여러 가지 변형이 가능함은 물론이다. 즉 본 발명은 통신 경로의 품질에 따라 압축되지 않은 패킷을 전송하는 주기를 변경하는 것에 관련된 것으로서, 패킷의 구성이나 패킷을 압축하는 방식이나 통신 경로의 품질을 측정하는 방식 등에 의해 제한되는 것은 아니다. 예를 들어 통신 경로의 품질은 노드간을 연결하는 경로의 일부인 무선 링크의 품질(신호대 간섭비 등), 라우터들의 부하율 등을 이용하여 측정될 수도 있다. 게다가 본 발명은 패킷의 헤더를 압축하여 전송하는 모든 종류의 프로토콜에 적용될 수 있는 것으로서, 도 2 및 도 3에 나타난 패킷의 포맷에 한정되는 것이 아니다.

<78> 그러므로 본 발명의 범위는 설명된 실시예에 국한되지 않으며, 후술되는 특허청구의 범위뿐만 아니라 이 특허청구의 범위와 균등한 것들에 의해 정해져야 한다.

【발명의 효과】

<79> 이상에서 상세히 설명한 바와 같이 동작하는 본 발명에 있어서, 개시되는 발명중 대표적인 것에 의하여 얻어지는 효과를 간단히 설명하면 다음과 같다.

<80> 본 발명은 다른 패킷의 손실에 의한 패킷의 폐기를 줄일 수 있다. 게다가 별도의 역방향 응답(Ack) 없이 압축 패킷의 폐기를 예방하기 때문에, 시스템의 부하를 가중시키지 않으면서 헤더 압축 기술에 의한 전송효율을 보다 향상시킬 수 있다. 특히 적어도 일부는 무선 링크로 구성된 패킷 통신 네트워크에서 헤더 압축 기술의 효과를 최적화한다.

【특허청구범위】**【청구항 1】**

패킷 통신 네트워크에서 패킷 데이터를 송신하는 방법에 있어서,

주기적으로 네트워크 상태를 측정하고, 상기 측정 결과에 따라 패킷 전송 주기를 결정하는 과정과,

상기 결정된 전송 주기에서는 압축되지 않은 헤더를 가지는 풀 패킷들을 전송하고
상기 결정된 전송 주기가 아닌 동안에는 압축된 헤더를 가지는 압축 패킷들을 송신하는
과정을 포함하는 것을 특징으로 하는 상기 방법.

【청구항 2】

제 1 항에 있어서, 상기 패킷 전송 주기를 결정하는 과정은,

최근 네트워크 상태 측정 주기 동안에 전송한 패킷들의 개수에 대해 재전송된 패킷들의 개수의 비율로서 패킷 재전송율을 계산하는 단계와,

상기 계산된 패킷 재전송율에 따라 상기 패킷 전송 주기를 결정하는 단계를 포함하는 것을 특징으로 하는 상기 방법.

【청구항 3】

제 2 항에 있어서, 상기 네트워크 상태 측정 주기는, RTP(Real-time Transmission Protocol)의 상태정보 전송 주기와 동일한 것을 특징으로 하는 상기 방법.

【청구항 4】

제 2 항에 있어서, 상기 패킷 전송 주기를 결정하는 단계는,

상기 패킷 재전송율이 미리 정해지는 상위 임계값보다 크거나 같으면 미리 정해지는 제1 전송 주기로 결정하고, 상기 패킷 재전송율이 미리 정해지는 하위 임계값보다 작거나 같으면 상기 제1 전송 주기보다 큰 제2 전송 주기로 결정하며, 상기 패킷 재전송율이 상기 상위 임계값과 상기 하위 임계값 사이이면 상기 패킷 전송 주기를 상기 제1 전송 주기보다 크고 상기 제2 전송 주기보다 작은 제3 전송 주기로 결정하는 것을 특징으로 하는 상기 방법.

【청구항 5】

제 4 항에 있어서, 상기 제1 전송 주기는 '1'인 것을 특징으로 하는 상기 방법.

【청구항 6】

제 4 항에 있어서, 상기 제2 전송 주기는 상기 제3 전송 주기의 2배인 것을 특징으로 하는 상기 방법.

【청구항 7】

제 1 항에 있어서, 상기 패킷 전송 주기를 결정하는 과정은,

최근 네트워크 상태 측정 주기 동안에 전송한 패킷들의 개수에 대해 재전송된 패킷들의 개수의 비율로서 패킷 재전송율을 계산하는 단계와,

상기 패킷 재전송율을 이전 네트워크 상태 측정 주기들 동안에 계산된 패킷 재전송율들에 누적하여 평균 패킷 재전송율을 계산하는 단계와,

상기 계산된 평균 패킷 재전송율에 따라 상기 패킷 전송 주기를 결정하는 단계를 포함하는 것을 특징으로 하는 상기 방법.

【청구항 8】

제 7 항에 있어서, 상기 평균 패킷 재전송율은 하기의 <수학식 2>에 따라 계산되는 것을 특징으로 하는 상기 방법.

【수학식 2】 $AR(i) = (1 - \alpha) * AR(i-1) + \alpha * R$

여기서 상기 AR은 상기 평균 패킷 재전송율이고, 상기 i는 네트워크 상태 측정 주기를 식별하는 인덱스이고, 상기 R은 패킷 재전송율이고, 상기 α 는 0과 1 사이의 값을 가지도록 미리 정해지는 가중치임.

【청구항 9】

제 7 항에 있어서, 상기 네트워크 상태 측정 주기는, RTP(Real-time Transmission Protocol)의 상태정보 전송 주기와 동일한 것을 특징으로 하는 상기 방법.

【청구항 10】

제 7 항에 있어서, 상기 패킷 전송 주기를 결정하는 단계는,

상기 평균 패킷 재전송율이 미리 정해지는 상위 임계값보다 크거나 같으면 미리 정해지는 제1 전송 주기로 결정하고, 상기 평균 패킷 재전송율이 미리 정해지는 하위 임계값보다 작거나 같으면 상기 제1 전송 주기보다 큰 제2 전송 주기로 결정하며, 상기 평균 패킷 재전송율이 상기 상위 임계값과 상기 하위 임계값 사이이면 상기 패킷 전송 주기를 상기 제1 전송 주기보다 크고 상기 제2 전송 주기보다 작은 제3 전송 주기로 결정하는 것을 특징으로 하는 상기 방법.

【청구항 11】

제 10 항에 있어서, 상기 제1 전송 주기는 '1'인 것을 특징으로 하는 상기 방법.

【청구항 12】

제 10 항에 있어서, 상기 제2 전송 주기는 상기 제3 전송 주기의 2배인 것을 특징으로 하는 상기 방법.

【청구항 13】

패킷 통신 네트워크에서 패킷 데이터를 송신하는 장치에 있어서,

주기적으로 네트워크 상태를 측정하고, 상기 측정 결과에 따라 패킷 전송 주기를 결정하는 제어기와,

상기 결정된 전송 주기에서는 압축되지 않은 헤더를 가지는 풀 패킷들을 전송하고 상기 결정된 전송 주기가 아닌 동안에는 압축된 헤더를 가지는 압축 패킷들을 송신하는 송신기를 포함하는 것을 특징으로 하는 상기 장치.

【청구항 14】

제 13 항에 있어서, 상기 제어기는,

최근 네트워크 상태 측정 주기 동안에 전송한 패킷들의 개수에 대해 재전송된 패킷들의 개수의 비율로서 패킷 재전송율을 계산하고, 상기 계산된 패킷 재전송율에 따라 상기 패킷 전송 주기를 결정하는 것을 특징으로 하는 상기 장치.

【청구항 15】

제 14 항에 있어서, 상기 제어기는,

상기 패킷 재전송율이 미리 정해지는 상위 임계값보다 크거나 같으면 미리 정해지는 제1 전송 주기로 결정하고, 상기 패킷 재전송율이 미리 정해지는 하위 임계값보다 작거나 같으면 상기 제1 전송 주기보다 큰 제2 전송 주기로 결정하며, 상기 패킷 재전송율이 상기 상위 임계값과 상기 하위 임계값 사이이면 상기 패킷 전송 주기를 상기 제1 전송 주기보다 크고 상기 제2 전송 주기보다 작은 제3 전송 주기로 결정하는 것을 특징으로 하는 상기 장치.

【청구항 16】

제 15 항에 있어서, 상기 제1 전송 주기는 '1'인 것을 특징으로 하는 상기 방법.

【청구항 17】

제 15 항에 있어서, 상기 제2 전송 주기는 상기 제3 전송 주기의 2배인 것을 특징으로 하는 상기 방법.

【청구항 18】

제 13 항에 있어서, 상기 제어기는,

최근 네트워크 상태 측정 주기 동안에 전송한 패킷들의 개수에 대해 재전송된 패킷들의 개수의 비율로서 패킷 재전송율을 계산하고, 상기 패킷 재전송율을 이전 네트워크 상태 측정 주기들 동안에 계산된 패킷 재전송율들에 누적하여 평균 패킷 재전송율을 계산하며, 상기 계산된 평균 패킷 재전송율에 따라 상기 패킷 전송 주기를 결정하는 것을 특징으로 하는 상기 장치.

【청구항 19】

제 18 항에 있어서, 상기 평균 패킷 재전송율은 하기의 <수학식 3>에 따라 계산되는 것을 특징으로 하는 상기 장치.

【수학식 3】
$$AR(i) = (1 - \alpha) * AR(i-1) + \alpha * R$$

여기서 상기 AR은 상기 평균 패킷 재전송율이고, 상기 i 는 네트워크 상태 측정 주기를 식별하는 인덱스이고, 상기 R 은 패킷 재전송율이고, 상기 α 는 0과 1 사이의 값을 가지도록 미리 정해지는 가중치임.

【청구항 20】

제 18 항에 있어서, 상기 제어기는,

상기 평균 패킷 재전송율이 미리 정해지는 상위 임계값보다 크거나 같으면 미리 정해지는 제1 전송 주기로 결정하고, 상기 평균 패킷 재전송율이 미리 정해지는 하위 임계값보다 작거나 같으면 상기 제1 전송 주기보다 큰 제2 전송 주기로 결정하며, 상기 평균 패킷 재전송율이 상기 상위 임계값과 상기 하위 임계값 사이이면 상기 패킷 전송 주기를 상기 제1 전송 주기보다 크고 상기 제2 전송 주기보다 작은 제3 전송 주기로 결정하는 것을 특징으로 하는 상기 장치.

【청구항 21】

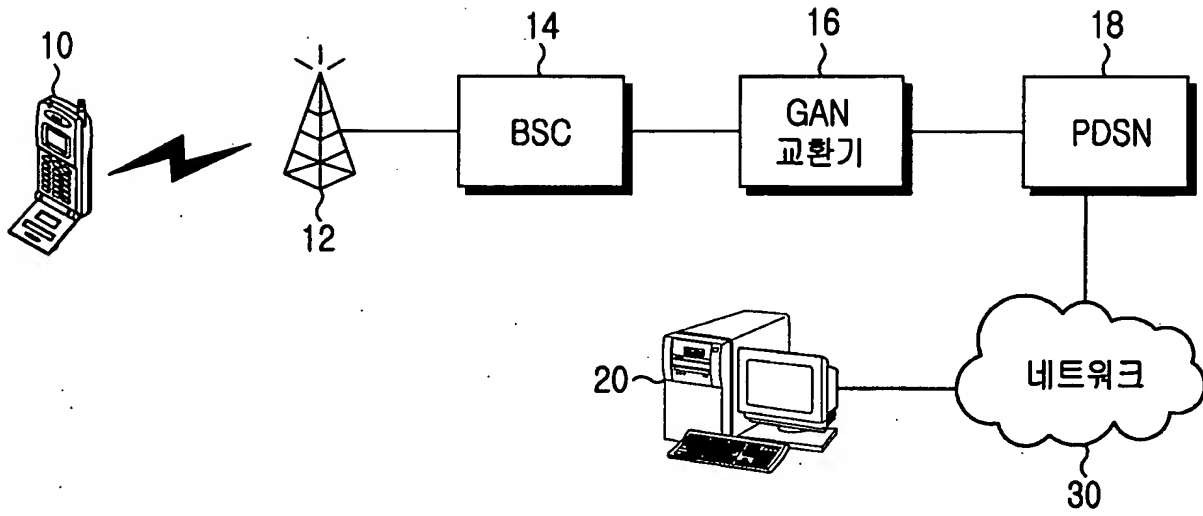
제 20 항에 있어서, 상기 제1 전송 주기는 '1'인 것을 특징으로 하는 상기 장치.

【청구항 22】

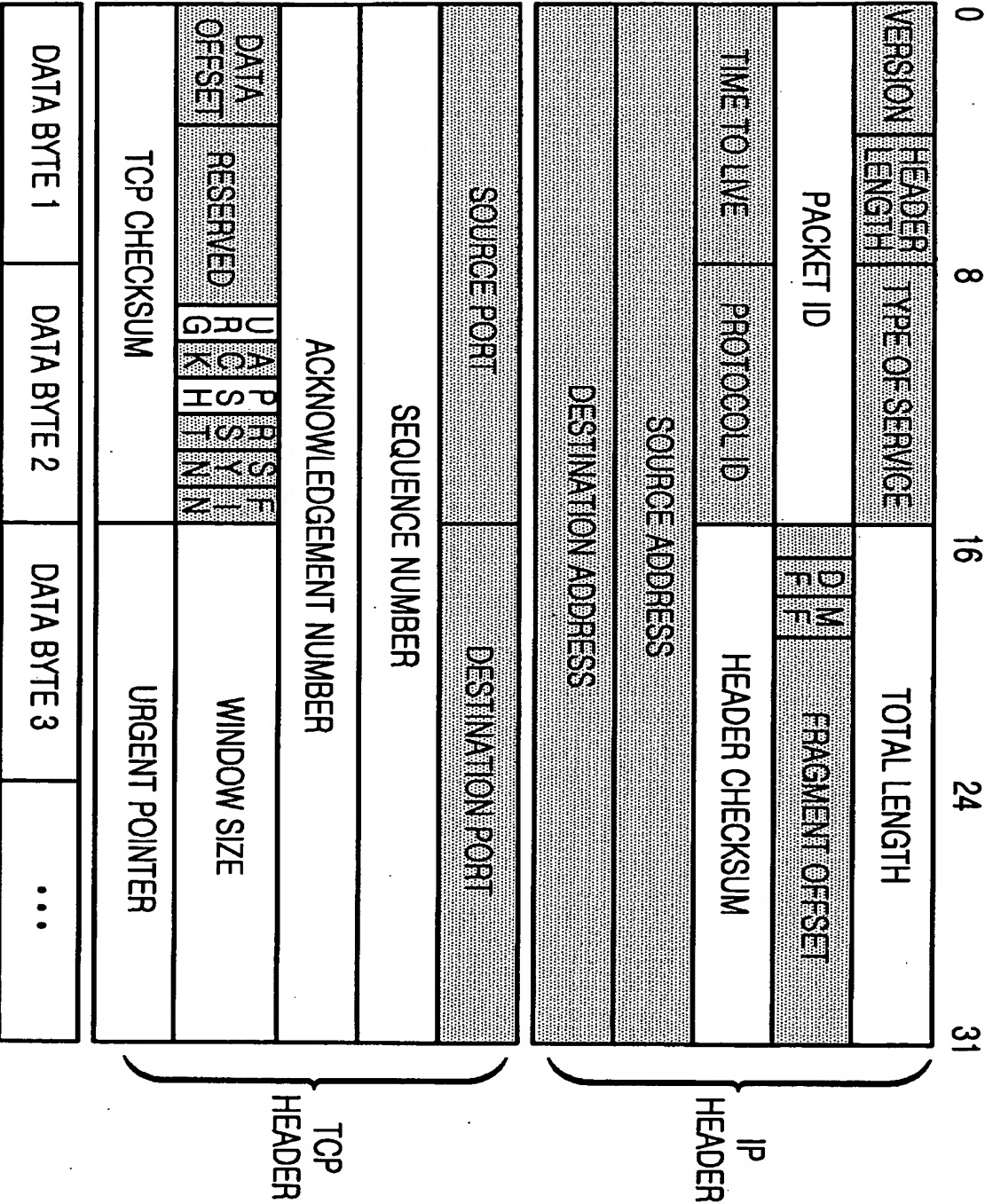
제 20 항에 있어서, 상기 제2 전송 주기는 상기 제3 전송 주기의 2배인 것을 특징으로 하는 상기 장치.

【도면】

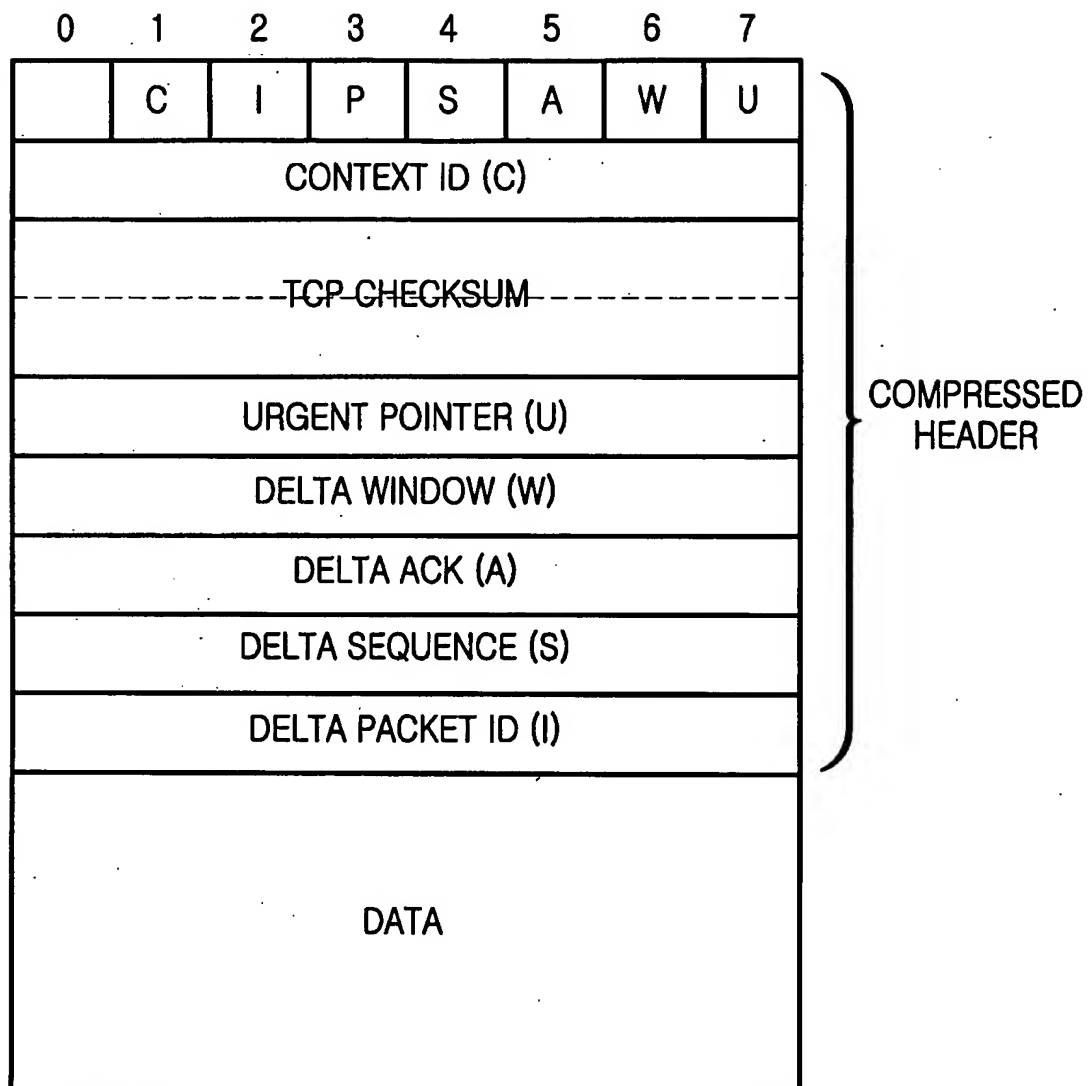
【도 1】



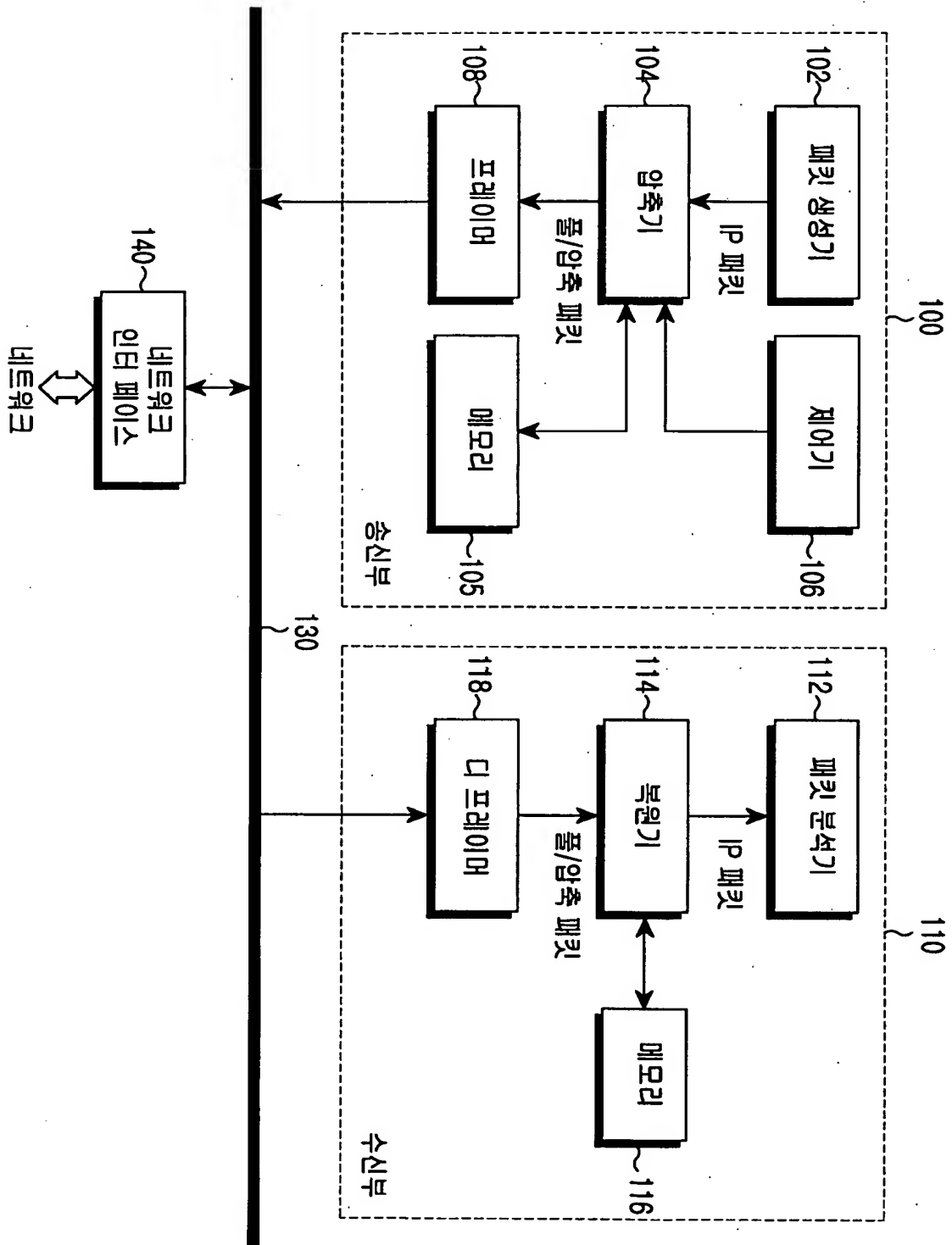
【도 2】



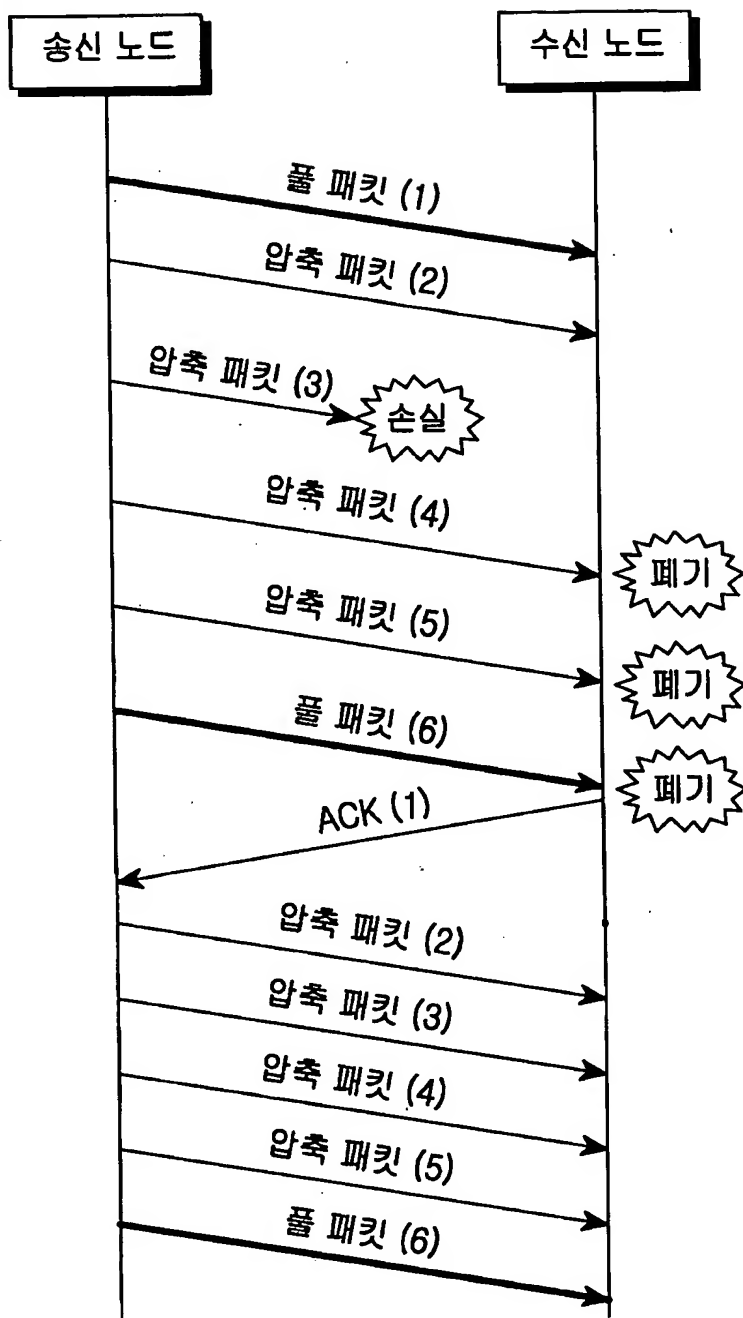
【도 3】



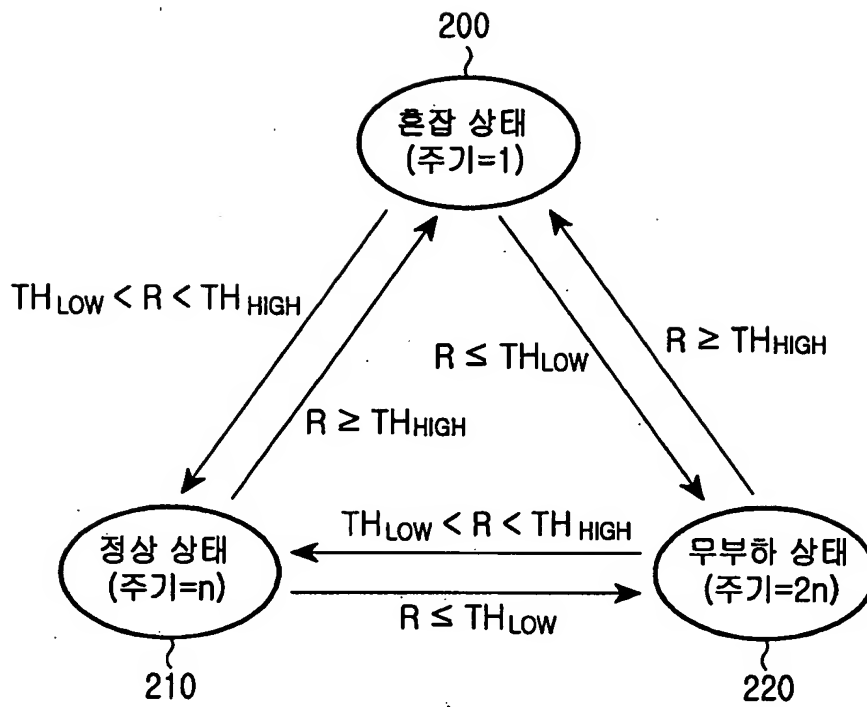
【도 4】



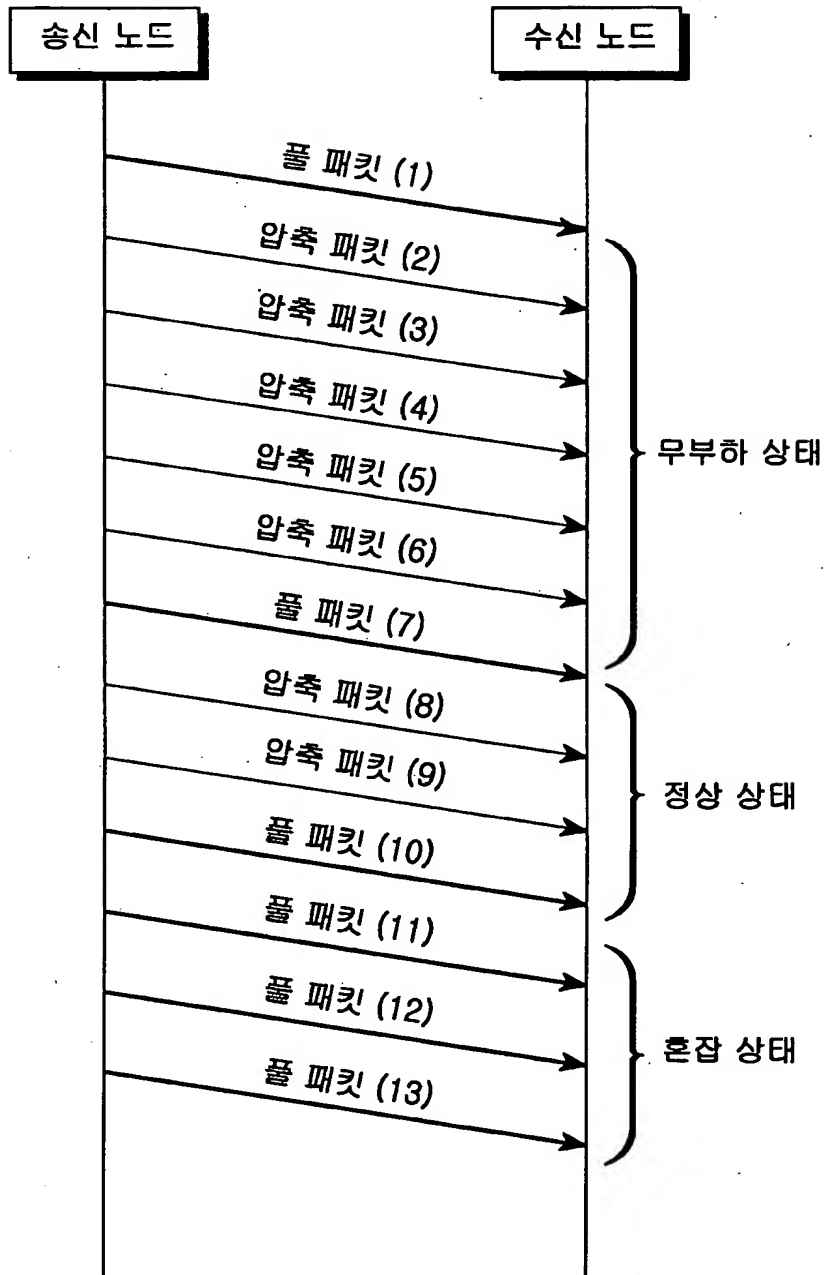
【도 5】



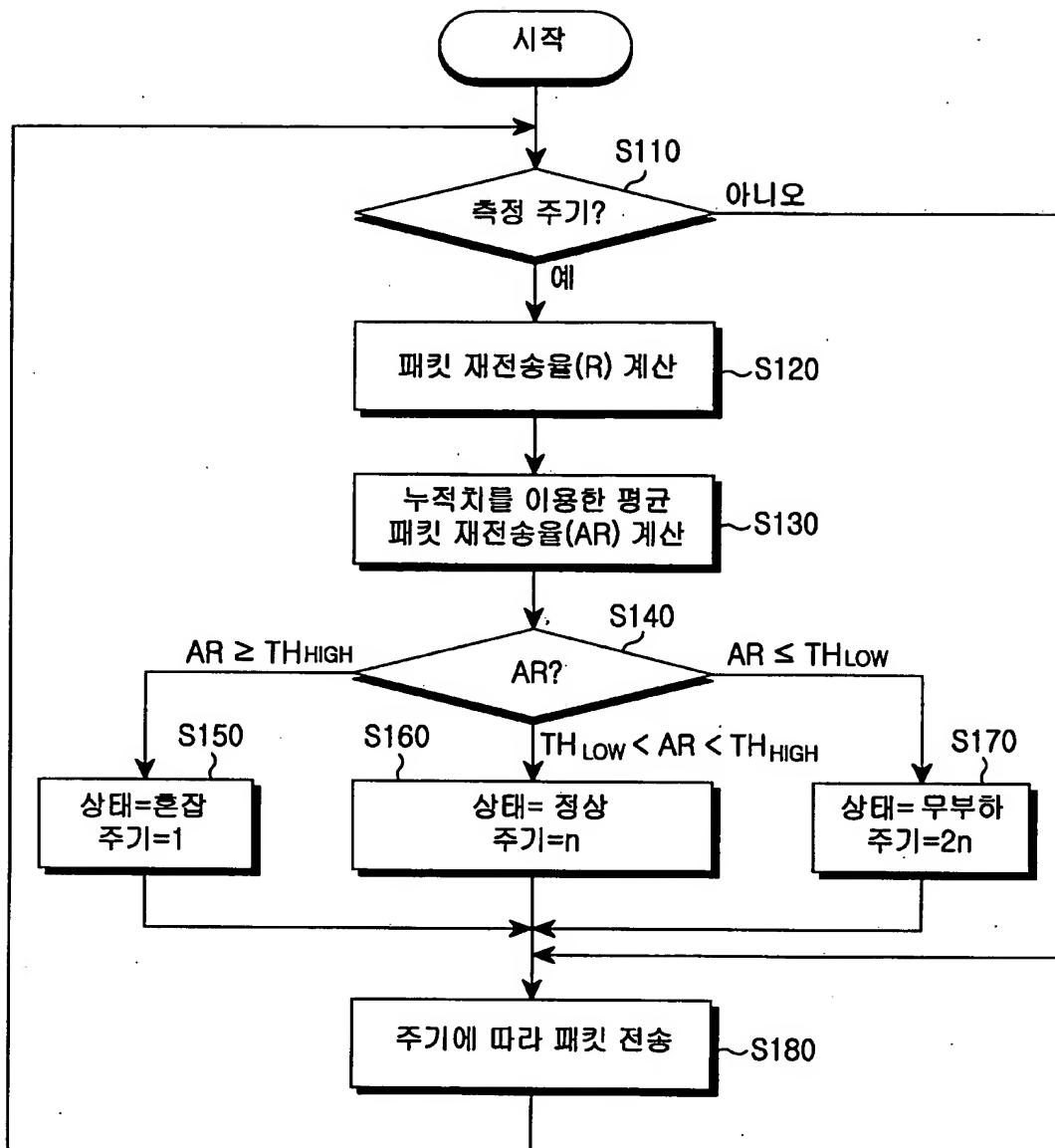
【도 6】



【도 7】



【도 8】



【도 9】

```

#define ALPHA 0.75
#define THHIGH 0.10
#define THLOW 0.05

// 5초 동안 발생한 패킷재전송율(재전송패킷/보낸 패킷수)을 계산한다.
(S120)    R = RetransPkts / SentPkts ;

// Low-pass-filter 공식을 사용하여 누적된 평균 패킷 재전송율을 계산한다.
(S130)    AR = (1-ALPHA)*AR
          +ALPHA * R;

// 패킷 재전송율에 따라 세가지 상태로 분류하고 각 상태에
// 따라 압축되지 않은 패킷을 언제 보낼지 결정한다.
if (AR >= THHIGH) {
(S150)    State = Congested;
          Period = 1;
} else if (AvgRetransRate <= THLOW) {
(S170)    State =unloaded;
          Period = 2n;
} else
(S160)    State = Loaded;
          Period =n;
}

// Period번째 마다 압축되지 않은 full packet을 보낸다.
(S180)    for (i=1; i<=Period; i++) {
          if (i == Period)
            send full packets;
          else
            send compressed packets;
        }

```